



Connecting you to the world
of **performance** & **capacity**.

Performance & Capacity 2014 by CMG

Load Testing: Which Tool to Choose?

Alexander Podelko

alex.podelko@oracle.com

alexanderpodelko.com/blog

@apodelko

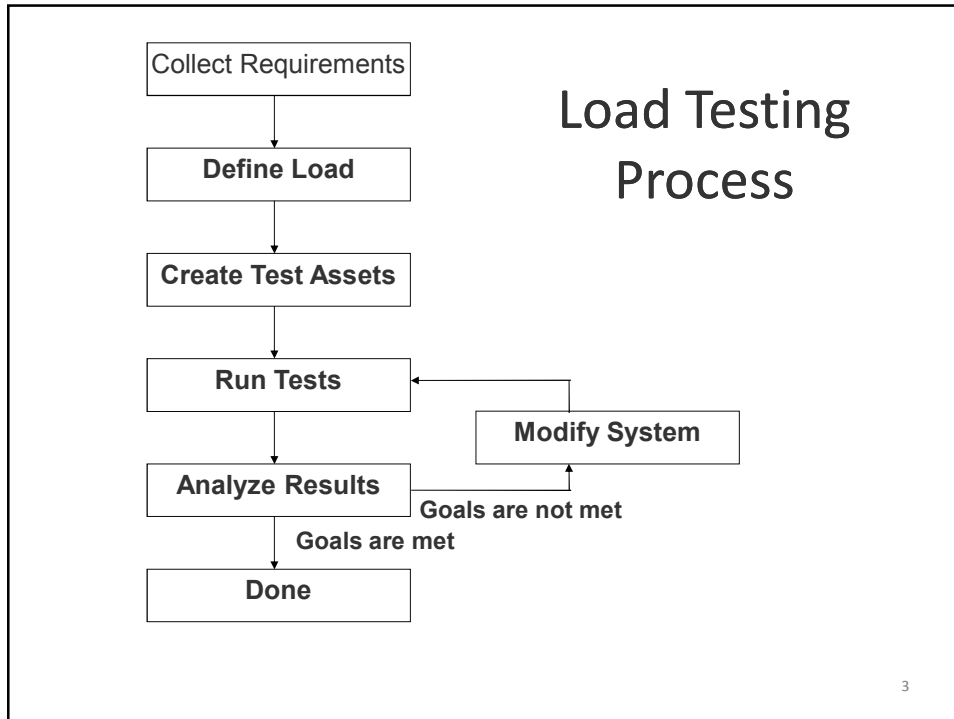
November 5, 2014

Agenda

- *Load Testing and Its Role*
- **Load Testing Tools**
 - **Technical Criteria**
 - **Non-technical Criteria**

Disclaimer: The views expressed here are my personal views only and do not necessarily represent those of my current or previous employers. All brands and trademarks mentioned are the property of their owners.

2



Load Testing

Applying multi-user synthetic load to the system

- Load testing
- Performance testing
- Stress testing
- Scalability testing
- Volume testing
- Reliability testing
- Concurrency testing
- Realistic Testing
- Endurance testing
- Longevity testing
- Soak testing
- Stability testing

4

The Stereotype

- Load / Performance Testing is:
 - Last moment before deployment
 - Last step in the waterfall process
 - Protocol level record-and-playback
 - Large corporations
 - Expensive tools requiring special skills
 - Lab environment
 - Scale-down environment
 - ...

5

Load Testing

- It is only one very specific kind of load testing
 - Most popular due to easier integration in corporate SDLC
- But load testing in no way limited to this one
 - Technology evaluation
 - Infrastructure evaluation
 - Prototypes / POC
 - Component / unit
 - What/if scenarios
 - Performance troubleshooting
 - Performance optimization
 - Benchmarking

6

Performance Risk Mitigation

- Single-user performance engineering
 - Profiling, WPO, single-user performance
- Software Performance Engineering
 - Modeling, Performance Patterns
- Instrumentation / APM / Monitoring
 - Production system insights
- Capacity Planning/Management
 - Resources Allocation
- Continuous Integration / Deployment
 - Ability to deploy and remove changes quickly

7

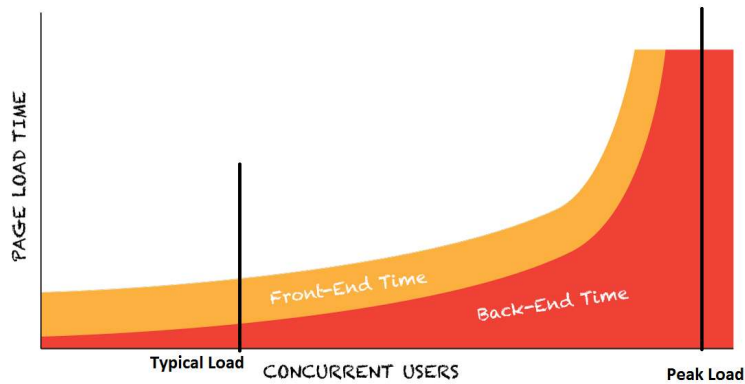
But all of them don't
replace load testing

**Load testing
complements them in
several important ways**

8

Can System Handle Peak Load?

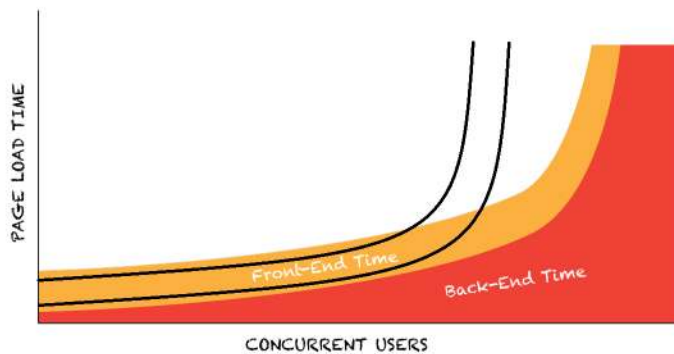
- You can't know without testing:



9

Verify Multi-User Performance

- Single-user improvement may lead to multi-user performance degradation



10

What Else Load Testing Adds

- Performance optimization
 - Apply exactly the same load
 - See if the change makes a difference
- Debugging/verification of multi-user issues
- Testing self-regulation functionality
 - Such as auto-scaling or changing the level of service depending on load

11

Agenda

- **Load Testing and Its Role**
- *Load Testing Tools*
 - **Technical Criteria**
 - **Non-technical Criteria**

12

Commercial Tools

Gartner Magic Quadrant for Quality Suites:

- Leaders:
 - HP (LoadRunner)
 - IBM (Rational Performance Testing)
 - Microsoft (Visual Studio Web Performance...)
 - Oracle (Application Testing Suite)
 - SOASTA(CloudTest)
 - Borland (SilkPerformer)
- Performance: Neotys (NeoLoad), Blazemeter

13

Open Source Tools

- 53 tools listed at
<http://www.opensourcetesting.org/performance.php>
- JMeter
 - Clear leader in popularity
- Gatling
 - Getting popularity recently
- Grinder, OpenSTA
 - Were popular/looks like no development recently

14

Mixed Business Models

- Advanced functionality
 - With small tests for free
- Freemium models
 - LoadRunner, NeoLoad (up 50 users free)
 - CloudTest (Lite up to 100 users free)
- BlazeMeter service using open source JMeter
- Important when more people get involved
 - Continuous Integration, Agile development

15

Load Testing Tools

Differ drastically in:

- Supported approaches / protocols
- Functionality / extensibility
- Supported environments
- Scalability
- Result analysis
- Integration (monitoring, APM, CI, etc.)
- Cost/Licensing
- Available support and skills

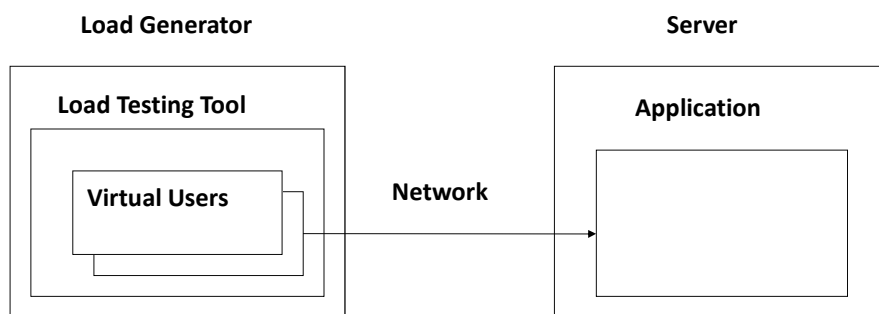
16

Agenda

- Load Testing and Its Role
- Load Testing Tools
 - Technical Criteria
 - Non-technical Criteria

17

Record and Playback: Protocol Level



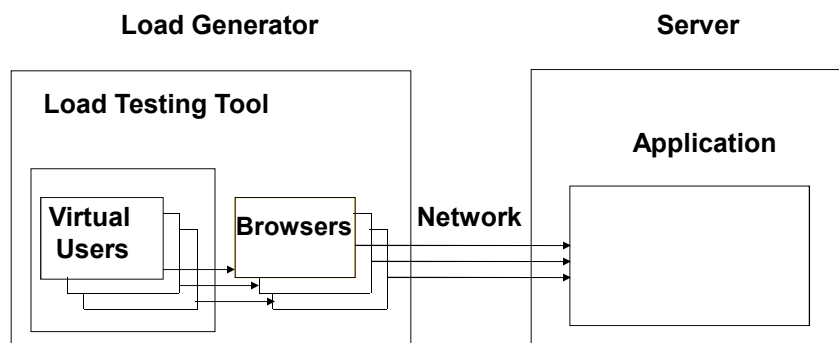
18

Considerations

- Usually doesn't work for testing components
- Each tool support a limited number of technologies (protocols)
- Some technologies are very time-consuming
- Workload validity in case of sophisticated logic on the client side is not guaranteed

19

Record and Playback: UI Level



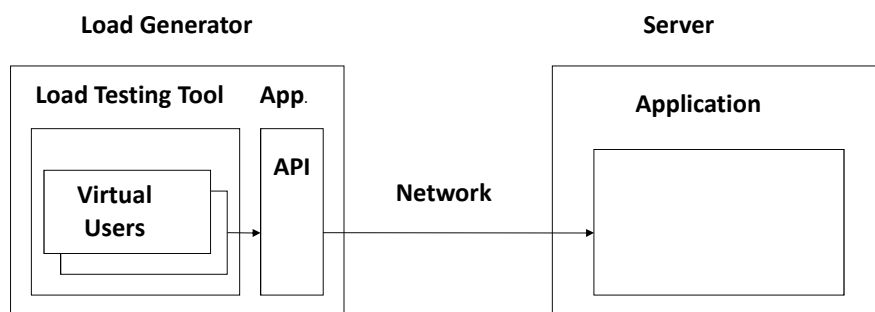
20

Considerations

- Scalability
 - Still require more resources
- Supported technologies
- Timing accuracy
- Playback accuracy
 - For example, for HtmlUnit

21

Programming



22

Considerations

- Requires programming / access to APIs
- Tool support
 - Extensibility
 - Language support
- May require more resources
- Environment may need to be set

23

Test Management

- Managing test execution / collecting results
- Virtual users coordination, e.g.:
 - Synchronization points
 - Data exchange
 - Sophisticated scheduling
- Environment simulation, e.g.:
 - Browser Simulation
 - Network simulation (including mobile)
 - IP spoofing

24

Deployment

- Lab vs. Service (SaaS) vs. Cloud (IaaS)
 - For both the system and load generators
- Test vs. Production
- No best solution, depends on your goals / system

25

Scenarios

- System validation for high load
 - Outside load (service or cloud), production system
 - Wider scope, lower repeatability
- Performance optimization / troubleshooting
 - Isolated lab environment
 - Limited scope, high repeatability
- Testing in Cloud
 - Lowering costs (in case of periodic tests)
 - Limited scope, low repeatability

26

Scaling

- Becomes **critical** as you get to a large number of virtual users
- The number of supported users per unit of computing power may differ drastically
 - Depending on tool, protocol, scenario, system...
- If you need deploy it on a large number of machines automation would be helpful

27

Reporting and Analysis

- Good integrated reporting and analysis greatly increases **efficiency**
 - Getting all data in one place and synchronized
 - Integration of monitoring data is a great help
- Weak spot of many open source tools

28

Monitoring

- System level
- Application level (APM)
 - AppDynamics, New Relic, Dynatrace, etc.
 - Many tools have integration
- Integration allows analyze monitoring data *together* with test results

29

Agile Support

- Agile / CI support becoming the main theme
- Integration with Continuous Integration Servers
 - Jenkins, Hudson, etc.
 - Several tools announced integration recently
 - Making a part of automatic build process
- Automation support
- Easiness to use
- Support of newest technologies

30

Agenda

- **Load Testing and Its Role**
- **Load Testing Tools**
 - **Technical Criteria**
 - Non-technical Criteria

31

Cost

- **Cost of licenses**
 - Usually not publicly available
 - Licensing restrictions are important
 - Free options: important for CI/Agile/etc
- **Cost of usage / efficiency**
 - Lack of functionality results in testers' time
 - [and, maybe, errors]

32

Support

- A lot of background sophistication
- Good support is important
 - Vendor support
 - Community support
 - Development support
 - New releases
 - Information support
 - Expert availability

33

Tool's Ecosphere

Name	Number of found documents by Google	Number of found US positions at Monster.com
[HP] LoadRunner	894,000	170
[Apache] JMeter	688,000	90
[Borland] SilkPerformer	138,000	12
[Neotys] NeoLoad	87,100	3
[SOASTA] CloudTest	51,500	2
LoadStorm	18,000	-

34

Summary

- ***There is no best tool*** – it depends on your needs
- Almost every tool works for a small and simple web site
- If using more sophisticated technologies, always check if the tool / approach supports it

35

Questions?

Alexander Podelko

alex.podelko@oracle.com
alexanderpodelko.com/blog
@apodelko